FingerTip Detection System Using Computer Vision In OpenCV

Rianna Maria Roshni Dsilva
Dept. of Information Technology
SRH Hochschule
Heidelberg, Germany
dsilva.rianna@gmail.com

***Abstract* – The aim of this project is to device a program that is able to detect fingertips, track them in real-time and to be further modeled as fingerprint recognition system for applications in biometrics. It is implemented using image processing using a regular laptop camera in OpenCV, Qt Creator and Visual Studio.**

Keywords – contour, segmentation, skin, region of interest, convex hull, defects, smoothing

INTRODUCTION

In the past few decades, human-beings have been addicted to various technologies such as captured photos, scanned signatures, bar code systems, RFID and so on. Biometrics is one of the applications in Image processing which refers to technologies that uses physiological or behavioral characteristics of the human body for user authentication. The biometric authentication system is based on two modes: Enrolment and Recognition. Biometric recognition is based on uniqueness and permanence. Uniqueness means that there is no similarity of features between two different biometrics data. For example, there are no two humans having the same fingerprint feature even if they are twins. When the features of biometrics do not change over the lifetime or aging, it is called permanence.

This paper presents the implementation and analysis of a real-time computer static vision, hand tracking system that can be used for interaction purposes in a mostly static environment with only the hands in the range of visibility using OpenCV – an open source library for computer vision and digital image processing. We apply background elimination using simple techniques and apply a set of filters to get the hand's contour, on which we create a convex hull and calculate the defect points in the hull which is used to detect the number of fingers, center and size of the palm.

The goal of this paper to obtain the fingertip images using the process steps as mentioned below;

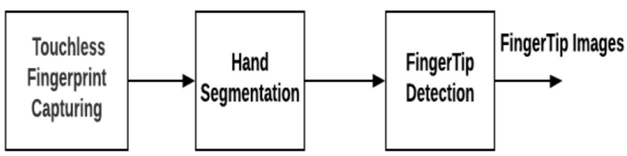


Figure 1: Methodology of Implementation

I. METHODOLOGY

A. Background Construction and subtraction

The first phase of the tracking system involves separating potential hand pixels from non-hand pixels. This can be performed by using Color Segmentation or Skin segmentation. Assume that you're in front of a background with a much different color from your skin color. Consider that your hand or palm is the largest part of your body being displayed on the camera. In this scenario, recognition of hand is simply done by selecting the pixels that belong to your skin color followed by retrieving the largest area.

B. Defining the Region of Interest

The region of interest in this project is the fingertip but in order to detect fingertip, it is necessary to first detect the hand. The captured frame is in the RGB (Red Blue Green) color space which is inverted to form the BGR. The BGR color space is converted to HSV (Hue Saturation Value). Hue indicates the color information, Saturation represents percentage of color and Value represents the brightness. This is done because the in the HSV color space the information of color is dissociated from the information of illumination. The range for human skin lies between [H = 0, S = 58] and [H = 50, S = 173]. A black and white image is obtained, where the pixels that have values within the interval are white, otherwise black.

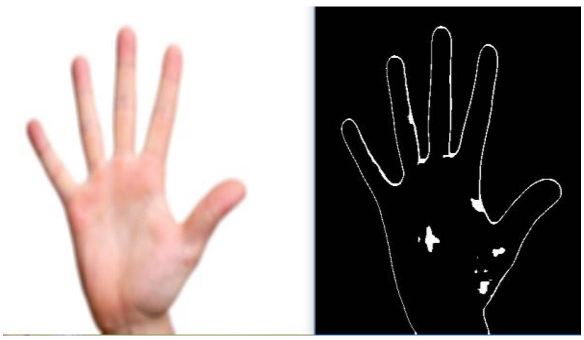


Figure 2: Before and After Hand Segmentation

C. Image Enhancement By Smoothing

The changes given by the background subtraction is very light and is high in noise. To suppress noise, we use a combination of techniques. These are morphological transformations. First is smoothing or blurring of an image using median filters. The median filter runs through each element of the image and replaces each pixel with the median of its neighboring pixels. OpenCV contains an in-built function for implementing Median Blur. Second technique is called Erosion or Dilation. Erosion erodes away the boundaries of foreground object. Dilation operations consists of convoluting an image with some kernel which can have any shape or size, usually a square or circle. The kernel has a defined anchor point, usually being the center of the kernel. As the kernel is scanned over the image, we compute the maximum pixel value overlapped by and replace the image pixel in the anchor point position with that maximal value. As you can deduce, this maximizing operation causes bright regions within an image to grow.

In the images below, we see the result of segmentation before the application of blurring, erosion and dilation. The obtained image is full of noise and is not ideal to obtain all fingerprint features. Hence, it is important to apply blurring, either Gaussian or Median and dilate or erode the image.

Figure 3: Four fingers after Segmentation

Figure 4: Five Fingers after Segmentation

Next, Median blurring and Dilation is applied, and the resultant image is shown below. The image seems much clearer than the original image after color space conversion. It is obvious that the image still contains irregularities. These irregularities can be fixed by a techniques called Contour Extraction.

Figure 5: Image after Erosion and Dilation

Figure 6 gives us a general idea of the Morphological techniques: Erosion (LHS) and Dilation (RHS).

Figure 6: Erosion and Dilation example

D. Contour Extraction

Contours can be explained simply as a curve joining all the continuous points along the boundary, having same color or intensity. The contours are a useful tool for shape analysis, object detection and recognition. For accuracy, image must be converted to binary image. Before finding contours, apply threshold or canny edge detection. OpenCV contains in-built functions to find and draw contours. In OpenCV, finding contours is like finding a white object from black background.

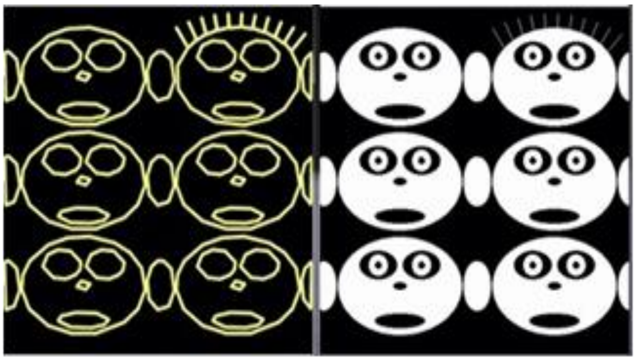


Figure 7: Image Contouring Example

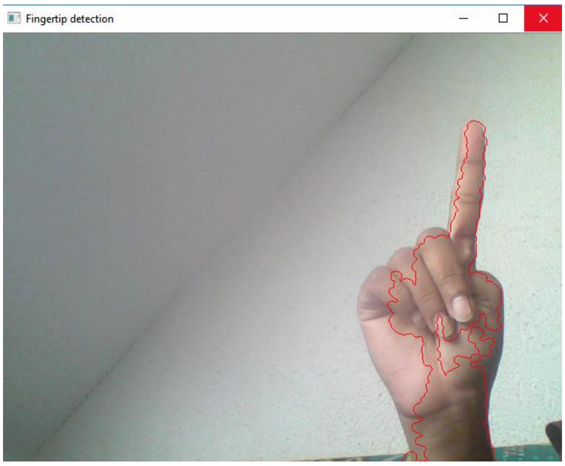


Figure 8: Contour of one finger

E. Convex Hull and Defects

In mathematics, Convex Hull is the smallest convex set that contains a set of points. And a convex set is a set of points such that, if we trace a straight line from any pair of points in the set, that line must be also be inside the region. The result is then a nice, smooth region, much easier to be analyzed than the contoured image, that contains many imperfections.

Given the set of points for the contour, we find the smallest area convex hull that covers the contours. The observation here is that the convex hull points are most likely to be on the fingers as they are the extremities and hence this fact can be used to detect number of fingers. There are "gaps" between the convex hull region and contour region. The "convexDefects" will try to approximate those gaps using straight lines. We can then use that information to find the points where our fingertips are placed.

However, there are far more points than just our fingertips. We need to do a filtering for only the points of our interest. We can think in some cheap but useful heuristics for that: i) Consider the inner angle between the two lines of the defect region to be between a certain interval; ii) Consider the angle between the initial point and the center of the contour region to be between a certain interval; iii) Consider the length of the line from the initial point to the middle point to be above a certain threshold.

The inner angle is exactly the angle between our fingers. Generally, the angle between our fingers is between 20° and 120°. The image below illustrates the concept;

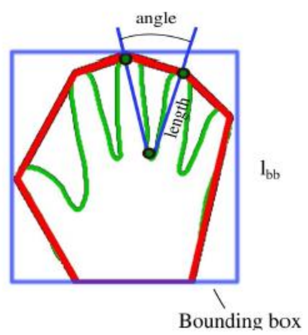


Figure 9: Creating a bounding box around palm

The angle between the initial point and the center of contour region is necessary to erase points located in the lower part of the contour. In order to find the center of the contour, we must involve it with a bounding box. Finally, the length of line from the initial point to the middle point can be calculated by simply calculating the Euclidean distance between the initial point and the middle point.

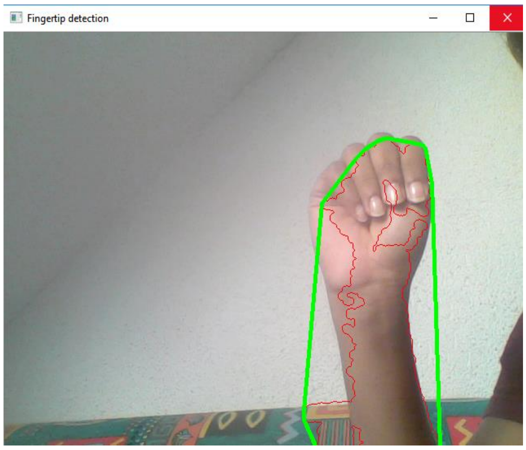


Figure 10: Bounding box around wrist

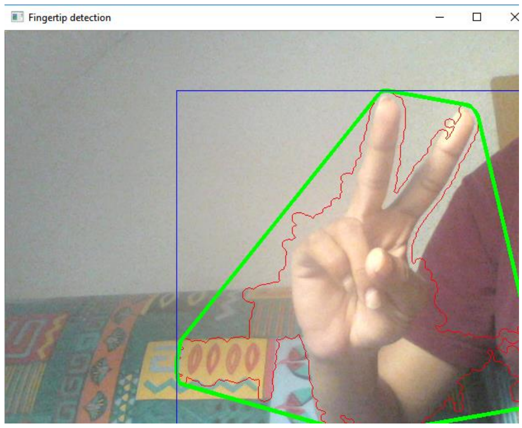


Figure 11: Bounding box around palm

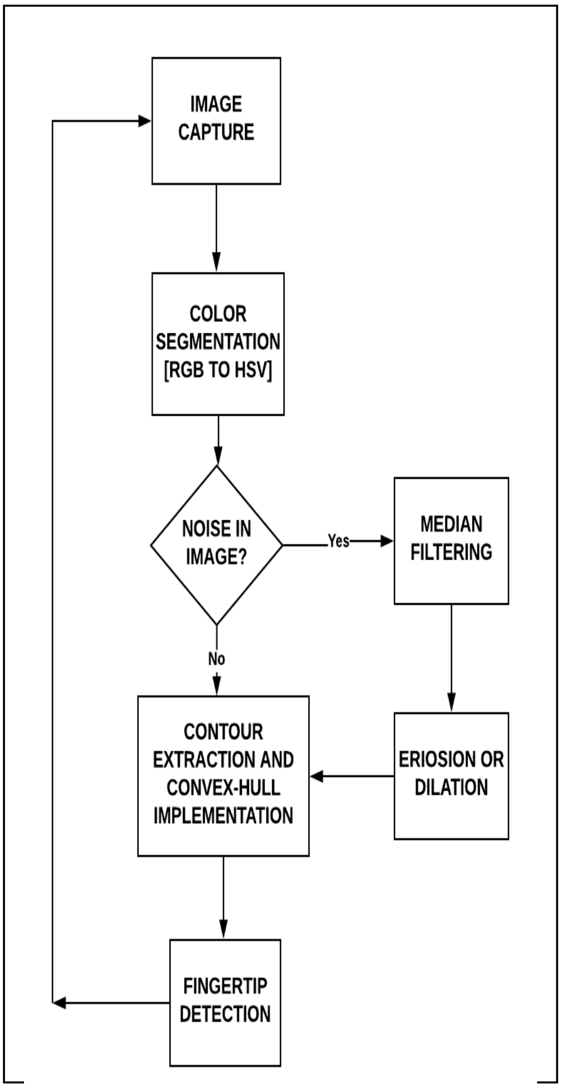
II. TECHNOLOGIES USED

III. IMPLEMENTATION CHALLENGES

Analysis of shortcomings and error types of current fingertip recognition systems identifies three principal challenges:

1. Background needs to be of a much different color from the individuals skin color.
2. Hand is positioned in front of the camera at a certain angle and rotation must be at a minimum as tilts cause the palm to result in an elliptical shape which is not well detected by the system.
3. Low quality laptop camera conditions that affect the performance of fingertip detection.
4. The background needs to be stationary. Slight to moderate motion is tolerable due to the filtering techniques used.

The flowchart below explains the technique used in this paper for fingertip detection.



The various stages are defined below:

Image Capture or Acquisition stage, Image pre-processing stage, defining region of interest, Image Enhancement, Morphological Techniques, Detect Fingertip.

IV. RESULT

To test the robustness of our framework various environments were chosen. More exactly, the experiments were conducted in rooms with different illumination settings and complex backgrounds. In order to increase the difficulty of the recognition task skin-like objects were deliberately chosen in the background scene. Also, the hand overlaps those objects. Considering the typical operating distance for an info-point application, the hand or palm are placed at a distance of 40 cm up to 120 cm from the camera.

The proposed algorithm was implemented using OpenCV 3.1.0 library and compiled using Visual Studio 2013 using Qt Creator as the integrated development environment. The proposed solution ran smoothly using Windows 10 and a Dell Inspiron 5458 notebook hardware platform based on Intel® Core™ i5-5200 CPU @ 2.20 Ghz, 8.00 GB RAM. The images were acquired with a HD 720P integrated webcam of the notebook.

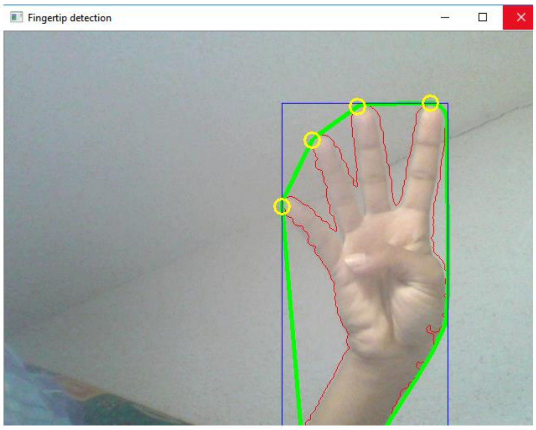
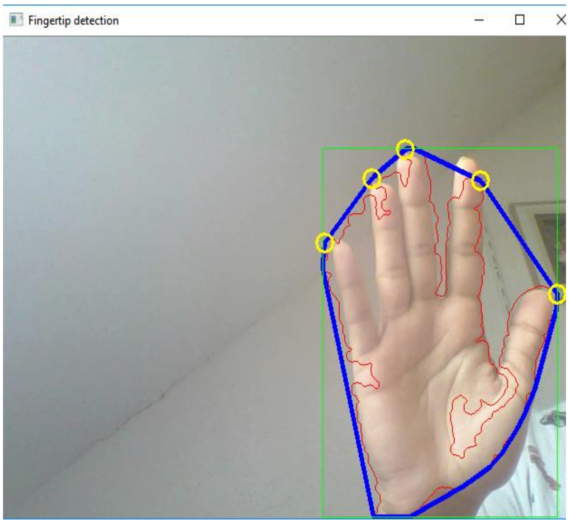


Figure 12: Fingertip Detected for four fingers

The results also showed that recognition application was quite robust for static images. However, the video version was enormously affected by the amount of illumination, such that is was necessary to check and adjust the HSV values for skin color when starting the program to get the proper output. Sometimes the adjustment was difficult to do because of the lighting conditions and the number of objects in the background. The application was very susceptible to noise on the video stream. Slight hand movements could affect gesture recognition. Nevertheless, if the hand is steady enough for long enough, the program outputs the correct command.

Figure 13: Fingertip Detection for five fingers



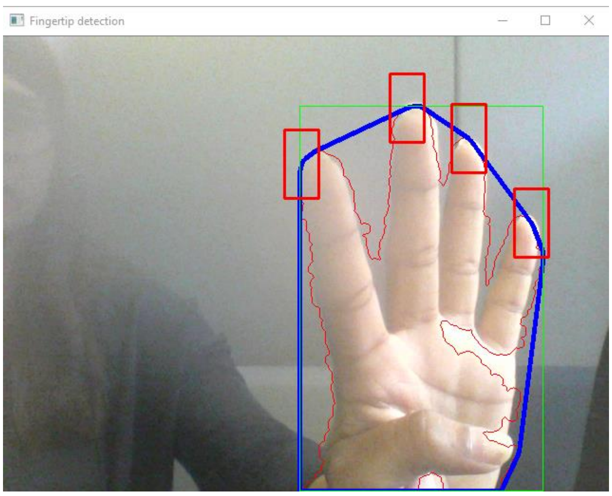


Figure 14: Fingertip Detection using rectangle

V. Applications in Real-Time

The fingertip detection system explained in this paper can be further developed into a fingerprint recognition system. This system can then be used for Biometric verification by comparing with stored fingerprint data in a database.

Some other applications are given as follows: Border control in Airports, Consumer or Residential biometrics, Financial applications, Time and attendance recording and maintaining, logical access control, justice and law enforcement, healthcare biometrics, mobile biometrics, biometric locks.

CONCLUSION AND FUTURE WORK

Through this paper we arrive at a conclusion that, for the purpose of verifying the identity of an individual using a fingerprint or fingertip detection, the algorithm must pass through several stages including: Image pre-processing such as Segmentation, Color Space transformation, Morphological transformations like Median filtering and Dilation, Contouring, Convex Hull techniques. Reasonable accuracy and stability is obtained which can be used further to implement a fingerprint biometric system for security purposes.

References

1. Christian von Hardenberg, (2001). Bare Hand Human-Computer Interaction. ACM International Conference Proceedings Series, Vol 15 , pp 1-8
2. Kenji Oka, Yoichi Sato and Jideki Koike, (2002). Real-Time Tracking of Multiple Fingertips and Gesture Recognition for Augmented Desk Interface Systems. Fifth IEEE International Conference on Automatic Face and Gesture Recognition.
3. Georgiana Simion, Ciprian David, Vasile Gui (2016) Fingertip-based Real Time Tracking and Gesture Recognition for Natural User Interfaces.
4. M. Corey, F. Farzam and J.H. Chong, The effect of linearization of range in skin detection, in: Proceedings of IEEE International Conference on Information, Communications and Signal Processing, Singapore, 10-13 December 2007.
5. U.A. Khan, M.I. Cheema and N.M. Sheikh, Adaptive video encoding based on skin tone

region detection, in: Proceedings of IEEE Students Conference, Pakistan, 16-17 August 2002.

6. A. Abadpour and S. Kasaei, Pixel-based skin detection for pornography filtering, Iranian Journal of Electrical and Electronic Engineering.
7. R.L. Hsu, M. Abdel-Mottaleb and A.K. Jain, Face detection in color images, IEEE Trans. Pattern Analysis and Machine Intelligence.
8. V. Vezhnevets, V. Sazonov and A. Andreeva, A survey on pixel-based skin color detection techniques, in: Proc. Graphicon, Moscow, September 2003.
9. J.B. Martinkauppi, M.N. Soriano and M.H. Laaksonen, Behavior of skin color under varying illumination seen by different cameras at different color spaces, in: Proc. Of SPIE, Machine Vision Applications in Industrial Inspection IX, USA, 2001.
10. A.M. Bazen and S.H. Gerez. Thin-Plate Spline Modelling of Elastic Deformations in Fingerprints. In Proc. SPS 2002, pages 205–208, Leuven, Belgium, March 2002.
11. A.M. Bazen and R.N.J. Veldhuis. Likelihood Ratio-Based Biometric Verification. In Proc. ProRISC 2002, 13th Annual Workshop on Circuits, Systems and Signal Processing, Veldhoven, The Netherlands, November 2002.
12. B. Bhanu and X. Tan. A Triplet Based Approach for Indexing of Fingerprint Database for Identification. In J. Bigun and F. Smeraldi, editors, Proc. 3rd Int. Conf. Audio- and Video-Based Biometric Person Authentication (AVBPA 2001), pages 205–210, Halmstad, Sweden, June 2001.
13. A.M. Bazen and S.H. Gerez. Thin-Plate Spline Modelling of Elastic Deformations in Fingerprints. In Proc. SPS 2002, pages 205–208, Leuven, Belgium, March 2002.
14. A.M. Bazen and R.N.J. Veldhuis. Likelihood Ratio-Based Biometric Verification. In Proc. ProRISC 2002, 13th Annual Workshop on Circuits, Systems and Signal Processing, Veldhoven, The Netherlands, November 2002.
15. B. Bhanu and X. Tan. A Triplet Based Approach for Indexing of Fingerprint Database for Identification. In J. Bigun and F. Smeraldi, editors, Proc. 3rd Int. Conf. Audio- and Video-Based Biometric Person Authentication (AVBPA 2001), pages 205–210, Halmstad, Sweden, June 2001
16. Lakshminarayanan. Hand Tracking And Gesture Recognition
17. Finger Detection in OpenCV – [Online Document] https://picoledelimao.github.io/blog/2015/11/15/fingertip-detection-on-opencv/

    Cited : 10th July 2018 - Available